Amendments to the Specification:

On page 1, please amend paragraph beginning on line 4 as follows:

A1

The present application relates to application Serial Number 09/522,369 ~~SN _____, RAL9-99-0156,~~ filed the same day than the present invention and assigned to the assignee of the present invention.

On page 1, please amend paragraph beginning on line 13 as follows:

A2

Segmenting variable length frames into fixed length cells or reassembling variable length frames from fixed length cells is handled today on network node adapters. The conversion from variable length frame to fixed length cells is commonly called segmenting even ~~either~~ if the frame length is longer or smaller than the cell length. Similarly, reassembly is used when one ~~used~~ uses fixed length cells to rebuild the variable length frame whatever the size of the cells compared to the size of the frame. If the frame segment length is smaller than the cell size, two frames segments can be packed into one cell. This will save bandwidth since it prevents sending a cell that is not entirely filled with frame data.

On page 2, please amend paragraph beginning on line 1 as follows:

A3

For this invention, the variable length frames can be any length and any protocol (Ethernet, Token Ring, or others). The fixed length cells are 4 column by 16 row cells (64 units total) where units can be either in bits, bytes or any multiplier of bits or bytes. For simplicity, bytes will be used ~~for~~ from now on. Each cell will have a 6 byte cell header. An optional 10 byte frame header can also be used after the cell header of the first cell. The contents and format of the cell and frame header and frame segment is not important for

A3

this invention. An example of a 64 byte fixed length cell is the cell length used with the IBM PRIZMA multiprotocol switch.

On page 2, please amend paragraph beginning on line 23 as follows:

A4

The background art with segmenting and reassembly comprises hardware oriented solutions using a non-negligible number of pointers and counters. To move the frame that has been received and buffered in the input buffer of a network equipment, pointers and counters are needed for keeping track of the data movement. Pointers are needed, for instance, to point to the offset in the input buffer of the next data to be moved to the output buffer and to point to the last data moved in the input buffer. Similarly, counters are needed to store the number of bits or bytes of the cell or the frame or the headers to be moved or already moved. The same needs for counters and pointers occurs when one wants to reassemble the frames. In ~~the~~ US patent 5,375,121, the flow chart of the cell assembly method illustrated in Fig.10 for converting network data into fixed length ATM cells, uses a significant number of pointers and counters. This has the inconvenience of using storage space and providing a complex process to coordinate all these counters.

On page 3, please amend paragraph beginning on line 12 as follows:

A5

The conversion from variable length frame to fixed length cells is commonly called segmenting either if the frame length is longer or smaller than the cell length. Similarly, reassembly is used when one used fixed length cells to rebuild the variable length frame whatever the size of the cells compared to the size of the frame. If the frame length is smaller than the cell size, the frame will be packed in cells.

On page 3, please amend paragraph beginning on line12 as follows:

a6

Particularly, if the apparatus is integrated on a chip, the greater the number of pointers and counters ~~is~~, the more space for gates (conditional logic and registers) and electrical power are needed on that chip. It is well known that for integration on a chip, the electrical power needed and the number of gates must be limited. Network chip manufacturers have thus to avoid such disadvantages to make competitive components.

On page 4, please insert the following new paragraph after line 19:

a7

The simple finite state machine is able to repetitively build all the cells corresponding to a frame because a cell pattern always applies to these cells with only two constraints. The first is to have the cell size be 64 unit (4 column unit by 16 row unit where a unit can be a byte, bit or any multiplier of bits or bytes) with a 6 unit cell header and 10 unit frame header; the second is to have the insert and overlay field have to be even. This method and apparatus can be used whenever the frame are modified during the segmenting processing by replacing a field by a definite value or when a new field is inserted in the frame or even if frames are packed into one cell.

On page 6, please amend paragraph beginning on line 16 as follows:

a8

Figure 9 is a flow chart describing the filling up of the cells as illustrated in Fig. 8, according to the [[a]] second embodiment of the invention building cell overlay;

On page 6, please amend paragraph beginning on line 19 as follows:

Figure 10 illustrates the data movement when filling up the cells as illustrated in Fig.8, according to [[a]] the second embodiment of the invention building cell overlay;

a8

On page 7, please amend paragraph beginning on line 6 as follows:

Figure 13 illustrates the data movement when filling up the cells as illustrated in Fig.11, according to [[a]] the third embodiment of the invention where a field is inserted in a cell;

On page 7, please amend paragraph beginning on line 11 as follows:

Figure 15 is a flow chart describing the filling up of the cells as illustrated in Fig. 15, according to the [[a]] the fourth embodiment of the invention where frames are packed in the cells;

On page 9, please amend paragraph beginning on line 19 as follows:

a9

Fig. 2 shows the main logical components of the preferred embodiments. Upon the assumption that the data store has been filled up with frames received from the network and that at the same time control information has been stored in the control blocks for the cells and the frames by the other components of the adapter. We ~~take~~ make the assumption that the frame to be segmented is one of the frames of the queue which has been scheduled by a scheduler in the adapter.

On page 10, please amend paragraph beginning on line 1 as follows:

a 10

The frames are stored in a storage unit, the data store (106), by words of 16 bytes. For an example, a first frame can be a 72 byte frame, it will be stored with four 16 byte words and a last 16 byte word where only 8 bytes are used [[on]] of the 16 bytes. One second frame could be a 32 byte frame filling up two words of 16 bytes. The data sturcture is ~~more~~ explained in detail in the comments for Fig.3 later in this document.

On page 11, please amend paragraph beginning on line 15 as follows:

a 14

The cell assembler accesses the data store and the CCI and outputs a cell (210) with the successive 64 bytes for each cell resulting in the segmenting of a frame. The cell is built in 16 steps each with 4 bytes transferred from the cell assembler to the [[an]] output 16 wire (4 byte) bus (205). Once the 16 rows of the cell are completed, the cell is ready to be sent on the bus of the switching fabric by one other component of the adapter.

On page 18, please amend paragraph beginning on line 9 as follows:

a 12

Fig.8 illustrates the same 9 cell pattern (401-408) ~~than~~ as illustrated in Fig.4, result of the segmenting of a frame having at least 570 bytes by the cell assembler of a second preferred embodiment. However, the difference with the first preferred embodiment in Fig.4 is the fact that the cell assembler, reading in one indicator of the CCI that it has to apply one overlay on a predefined field of the frame, has applied it as an additional step in its processing. The result illustrated in Fig.8 (800) is that the fourth 4 byte word after the frame header in the frame has been replaced by a predefined 4 byte value which has been read from the data in the CCI and multiplexed on the multiplexer (211) of the cell assembler. The new value of the field in the cell can be a replacement of an address in an

a 12

Ethernet frame. This could be particularly done to replace VLAN header values. Whatever the place of the field in the frame or the value of the field replacing it, this does not change the fact that, as illustrated in Fig.8, the cell pattern is respected and the same cell assembly method is used.

On page 19, please amend paragraph beginning on line 6 as follows:

a 13

Fig.10 illustrates the data movement for each cell register in the case of overlay. Similarly to the data movement illustrated in Fig.6, the movement of data can be for building the start of frame cell (1001 and 1002) or for building a continuation of frame cell (1001). However, a new data movement occurs when the cell to be built is the one having the overlay (the first cell of the frame in the preferred embodiment illustrated in Fig. 9). In this case (1003) the bytes 0 to 3 (column C3:C0) of the R7 row of the first cell is replaced by the new predefined 4 byte data. As usual when filling up the row, the PIB counter is incremented with the number of bytes even if these bytes do not come from the data store. This means that 4 bytes are skipped in the data store. The data movement goes on as in Fig.6 restarting at the R8 row.

On page 24, please amend paragraph beginning on line 12 as follows:

a 14

The same preferred embodiments are used for forming the frame reassembly in an output adapter (111) receiving fixed length cells from the bidirectional bus of the switched fabric and having to rebuild the variable length frames before sending them on the target LAN output ports (103). The frame assembler of the output adapter will use CCI information provided by a cell reassembly process to rebuild the frame. The second inputs are the cells themselves. The CCI and cell inputs are multiplexed in a two entry multiplexer. The output bus of the multiplexer is for transferring the frame data to the data storage. A

finite state machine reading the information in the CCI and the cell data is able, according to the same cell pattern ~~than~~ as in the input adapter, to rebuilt the successive data storage words of the frame data. The finite state machine activates the multiplexer in such a way that the cell header and frame header are suppressed. As in the cell assembler of the input adapter, the frame reassembler maintains a unique counter initialized to zero at the beginning of the cell reading and maintains this counter with the address in the word of the data storage where the next cell data will be written. As with the cell assembler the counter is incremented modulo 16, the word length in the data storage. When using the preferred embodiments the flow chart of Fig.5 is used for frame reassembly for each row read from the cell filled up by the switch fabric. As in Fig.6, for each cell row, data can be stored in the data store until a 16 byte word is formed. A new 16 byte word is to be built as the cell pattern is known and the PIB value is known as explained in Fig.7. The frame stored in the data store as 16 byte words will be used by the output scheduler to send the frames as soon as they are ready. This scheduling being the equivalent process available in the input adapter also for reading frame and writing them by 16 byte words in the data store.

a 14